
程式操作手冊

操作環境：

程式開發使用系統：

- 01. CentOS 8.0
- 02. Arch Linux 2020

推薦適用系統：

以下按推薦度編排，推薦度高者由上而下

- 01. Linux（依賴套件和路徑配置簡易）
 - 02. Mac
 - 03. Windows（Python2和Python3的共存設定繁複、theano執行效率差）
-

安裝軟體：

- Python （建議使用 Anaconda Python，在安裝依賴時更為便利）
 - Python 2.7 （MLnet 模型所需，以別名 python2 執行）
 - Python 3.6 以上 （分析 Saliency 所需）

須同時存在 2 種版本

依賴套件：

H3 Python 2 環境相關

使用 Python 2 環境的原因：因為其中使用到的視覺顯著性分佈預測類神經網路模型 [MLnet](#)，其開發環境在 Python 2 環境上並使用 Theano + Keras 達成，故必須將 Python 2 環境設定好。

首先，在 **Python 2.7** 環境 中安裝以下依賴套件：

01. Theano 0.9.0

02. Keras 1.2.2

03. Python OpenCV 3.4.3

or

Python OpenCV 3.2.0

以下提供用 Anaconda Python 安裝以上依賴套件的方法：



```
conda install -c conda-forge theano=0.9.0
conda install -c conda-forge keras=1.0.7
conda install -c conda-forge opencv=3.4.3
```

Keras 安裝示意：

```

Anaconda Prompt (Anaconda3) - conda install -c conda-forge keras=1.0.7

==> WARNING: A newer version of conda exists. <==
current version: 4.9.2
latest version: 4.10.3

Please update conda by running

$ conda update -n base -c defaults conda

## Package Plan ##

environment location: G:\Anaconda3\envs\py27

added / updated specs:
- keras=1.0.7

The following packages will be downloaded:

package | build | size | channel
-----|-----|-----|-----
h5py-2.9.0 | nompi_py27he41fcd5_1102 | 880 KB | conda-forge
hdf5-1.10.4 | nompi_h08085ad_1106 | 18.4 MB | conda-forge
keras-1.0.7 | py27_0 | 232 KB | conda-forge
linecache2-1.0.0 | py_1 | 13 KB | conda-forge
pyreadline-2.1 | py27_1001 | 136 KB | conda-forge
pyyaml-5.3.1 | py27h462b5f4_0 | 150 KB | conda-forge
traceback2-1.4.0 | py27_0 | 28 KB | conda-forge
unittest2-1.1.0 | py_0 | 68 KB | conda-forge
yaml-0.2.2 | h0c8e037_1 | 50 KB | conda-forge
zlib-1.2.11 | h3cc03e0_1006 | 117 KB | conda-forge
-----|-----|-----|-----
Total: | | 20.0 MB |

The following NEW packages will be INSTALLED:

h5py | conda-forge/win-64::h5py-2.9.0-nompi_py27he41fcd5_1102
hdf5 | conda-forge/win-64::hdf5-1.10.4-nompi_h08085ad_1106
keras | conda-forge/win-64::keras-1.0.7-py27_0
linecache2 | conda-forge/noarch::linecache2-1.0.0-py_1
pyreadline | conda-forge/win-64::pyreadline-2.1-py27_1001
pyyaml | conda-forge/win-64::pyyaml-5.3.1-py27h462b5f4_0
traceback2 | conda-forge/noarch::traceback2-1.4.0-py27_0
unittest2 | conda-forge/noarch::unittest2-1.1.0-py_0
yaml | conda-forge/win-64::yaml-0.2.2-h0c8e037_1
zlib | conda-forge/win-64::zlib-1.2.11-h3cc03e0_1006

Proceed ([y]/n)? y

Downloading and Extracting Packages
unittest2-1.1.0 | 68 KB | ##### 100%
traceback2-1.4.0 | 28 KB | ##### 100%
pyreadline-2.1 | 136 KB | ##### 100%
pyyaml-5.3.1 | 150 KB | ##### 100%
yaml-0.2.2 | 50 KB | ##### 100%
keras-1.0.7 | 232 KB | ##### 100%
hdf5-1.10.4 | 18.4 MB | ##### 100%
linecache2-1.0.0 | 13 KB | ##### 100%
zlib-1.2.11 | 117 KB | ##### 100%
h5py-2.9.0 | 880 KB | ##### 100%
Preparing transaction: done
Verifying transaction: \

```

H3 Python 3 環境相關

Python 3 是省電系統開發的環境，而為了在 Python 3 環境下調用 Python 2 的類神經網路模型，推薦使用 Python 3.6 以上的版本，因其內建 `subprocesses.run()` 函式；若使用舊版，則必須自行安裝。

`subprocesses.run` 安裝方法：



```
pip install subprocess.run
```

以下則為省電色盤系統開發時所使用的套件：

01. ColorThief



```
pip install colorthief
```

02. Pillow



```
pip install Pillow
```

03. pandas



```
pip install pandas
```

04. numpy



```
pip install numpy
```

05. elementpath



```
pip install elementpath
```

程式執行

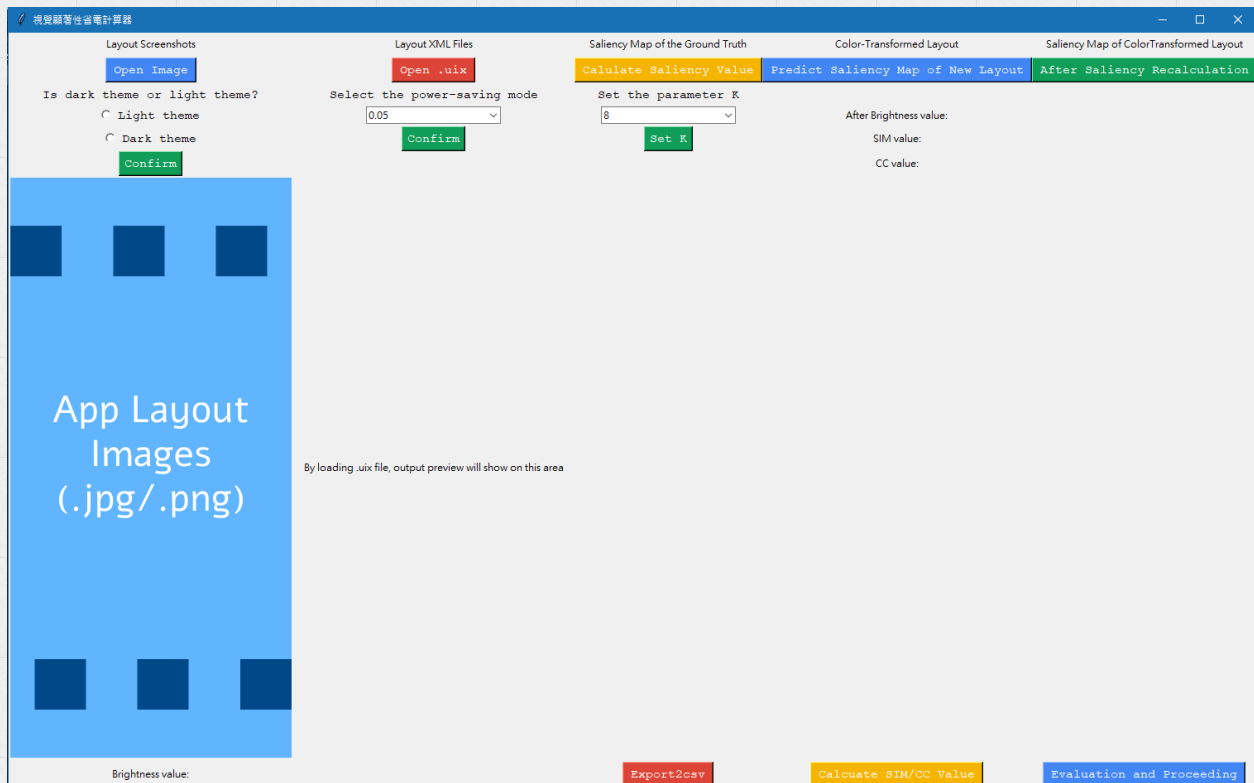
由於本程式涉及 Python 2 及 Python 3 環境的相互呼叫，打包後的執行程式檔極其容易出錯（目前僅成功於原開發機CentOS 8.0系統完美執行，於其他主機均會報錯），故以下操作均為對程式原始碼的運行（若以上套件安裝完成，此步驟於 Linux、MacOS及Windows都能有效執行）。

H3 基本操作方式

於各編輯器中，開啓「tkSaliencyMain.py」，並按下「run」或「debug」。

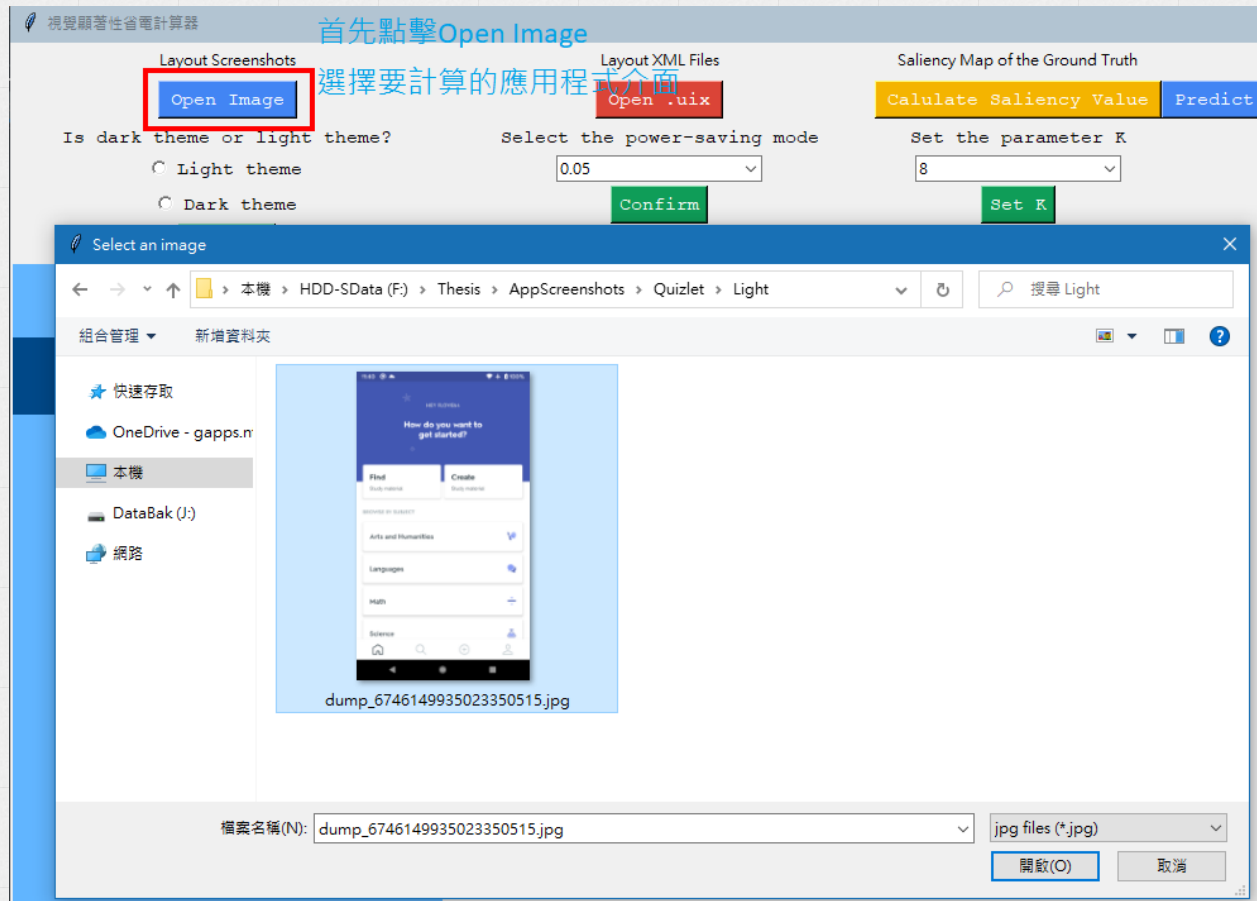
以 Spyder 編輯器示範圖如下：

接著會看到極其陽春的程式介面浮現：

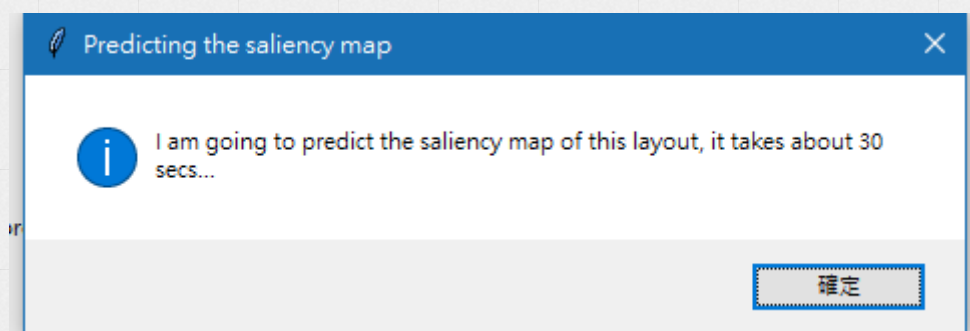


H3 預測介面圖片的視覺顯著性分佈

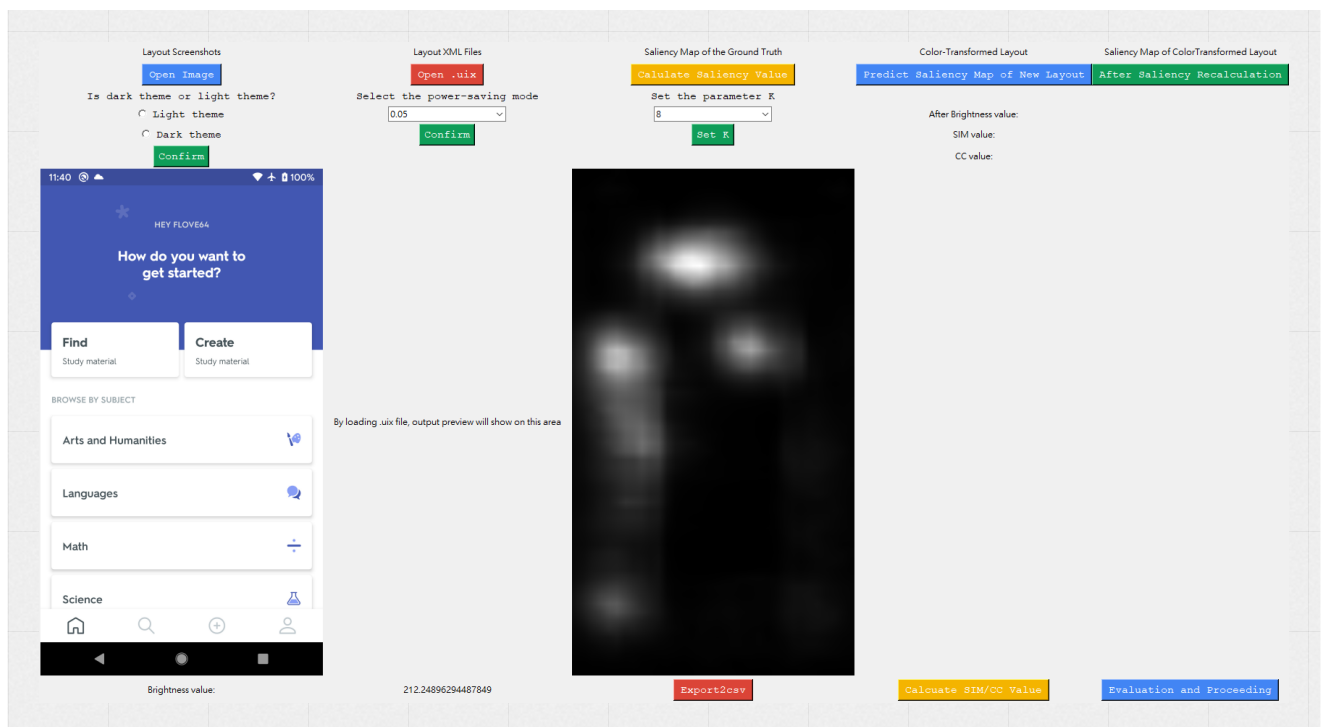
點擊左上角的 Open Image，選擇要進行預測的程式介面截圖，推薦使用 .jpg 格式，Python 對 .jpg 格式的圖像支援最為全面：



出現「等待提示」視窗，點擊確定，等待30秒左右即可看到預測出來的視覺顯著性分佈圖：



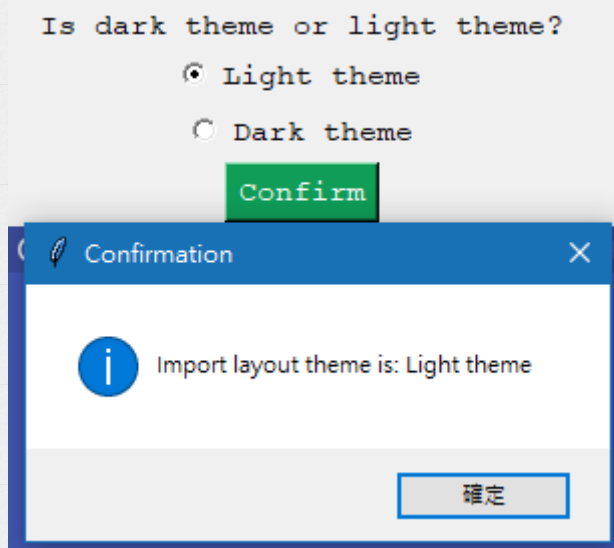
接著即預測出該介面的視覺顯著性分佈：



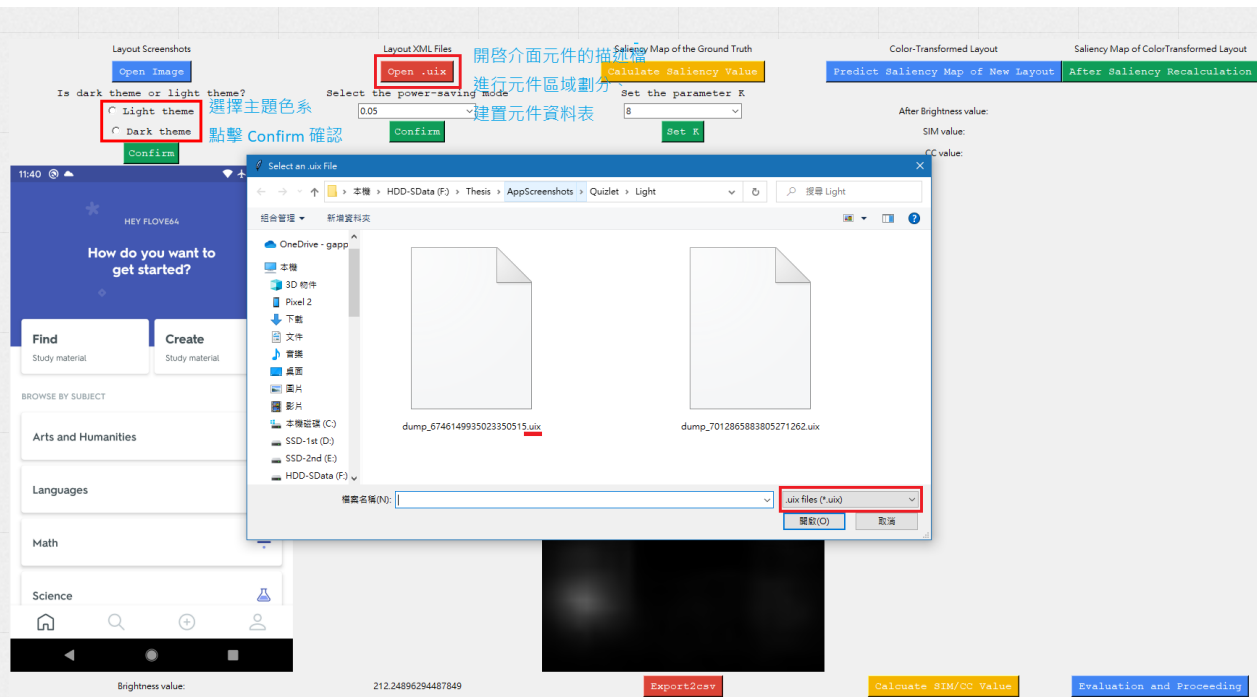
備註：如果此步驟無回應，極可能是系統並沒有識別到 Python 2 環境，或是 Python 2 環境中的依賴套件未安裝好所致。

H3 劃分應用程式元件區域

選擇主題色系（明亮主題、暗沉主題）並點擊確認：

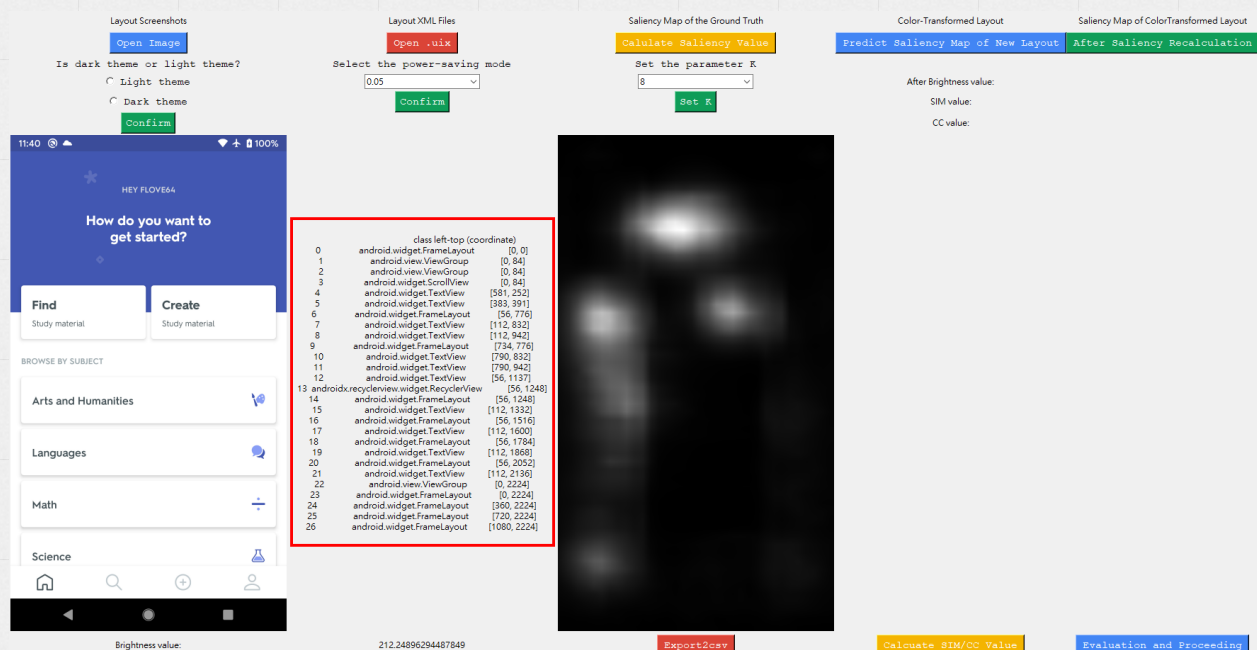


讀取介面元件描述檔（.uix）：



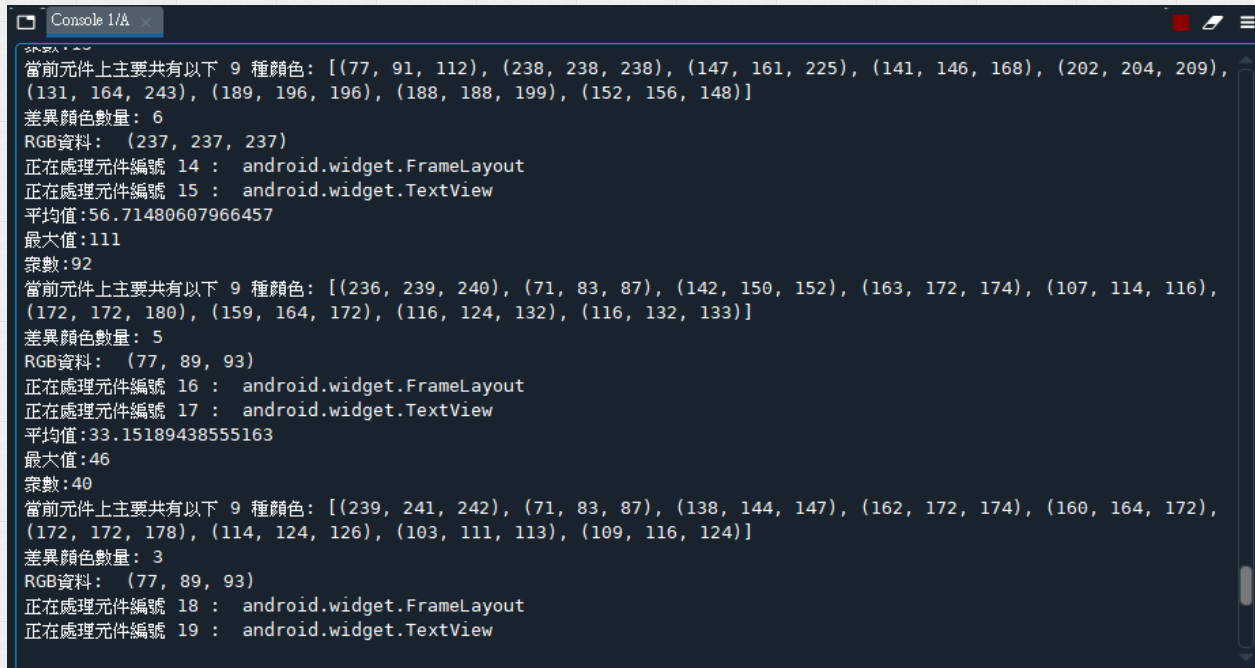
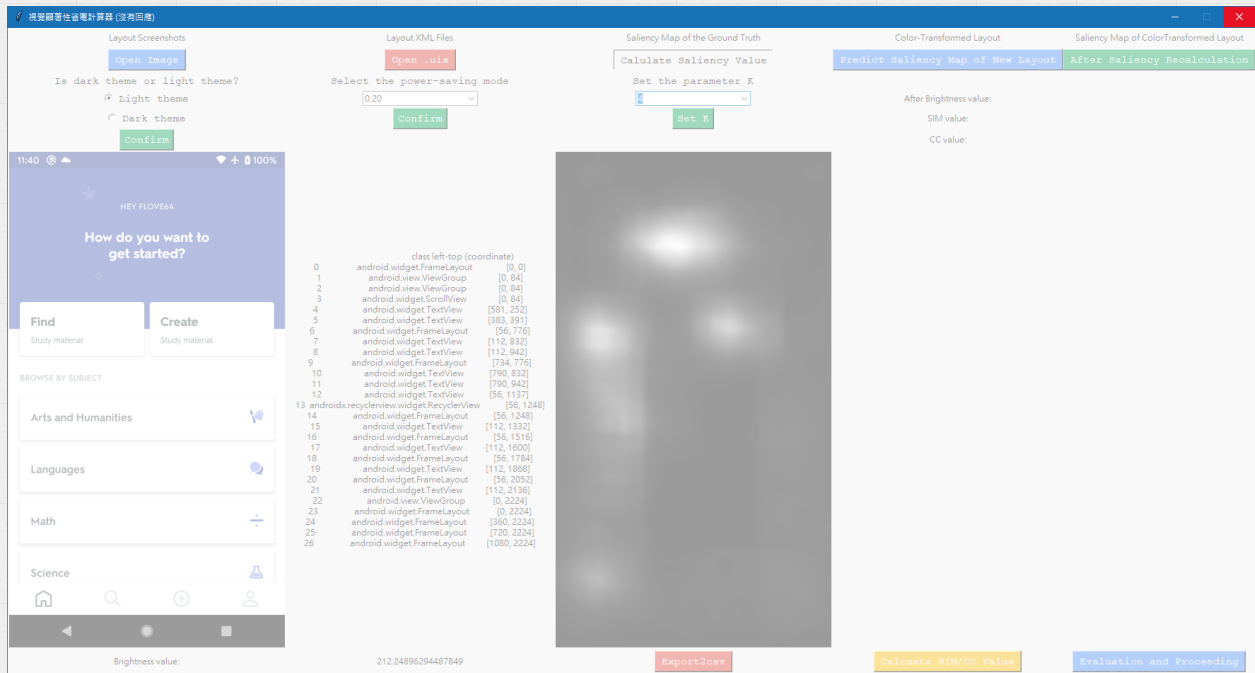
備註：介面元件描述檔（.uix）取自於 Google 的 uiautomator 工具，該工具被打包於 Android SDK 中，使用該工具對應用程式進行截圖時，便會取得該檔案（.uix）。

元件資料表便會顯示於介面中：

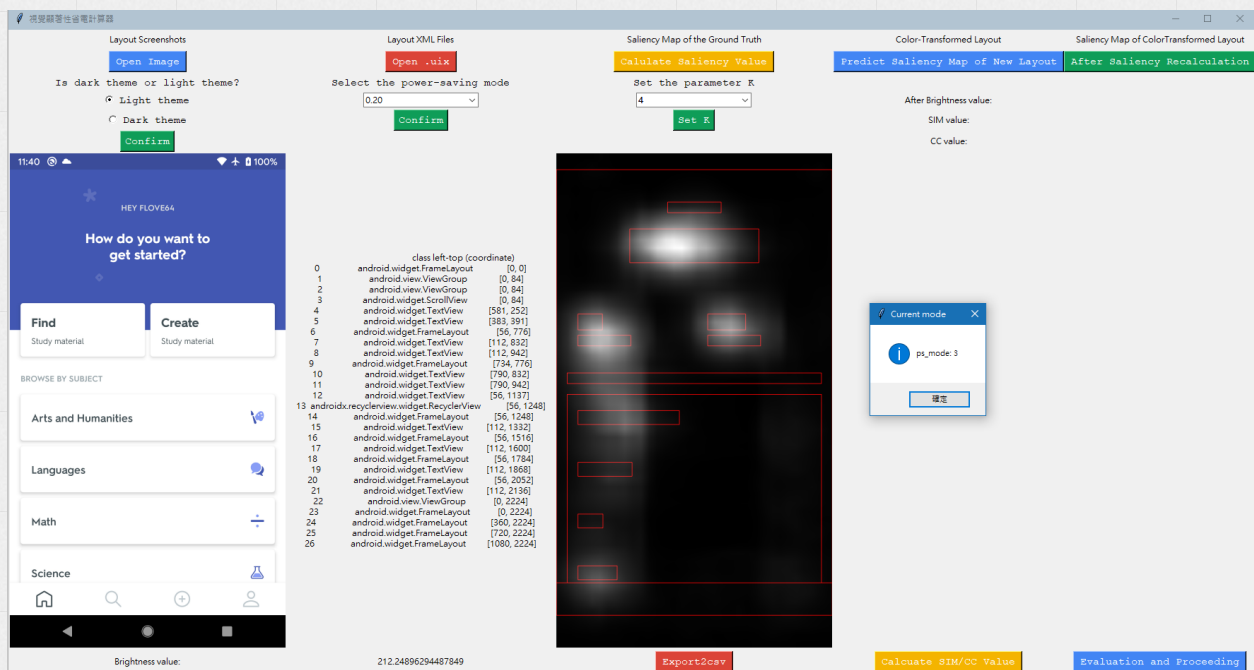


選擇容許的視覺顯著性差異目標參數及矚目元件參數K，即可輸出元件劃分區域，並且取得初始省電介面。

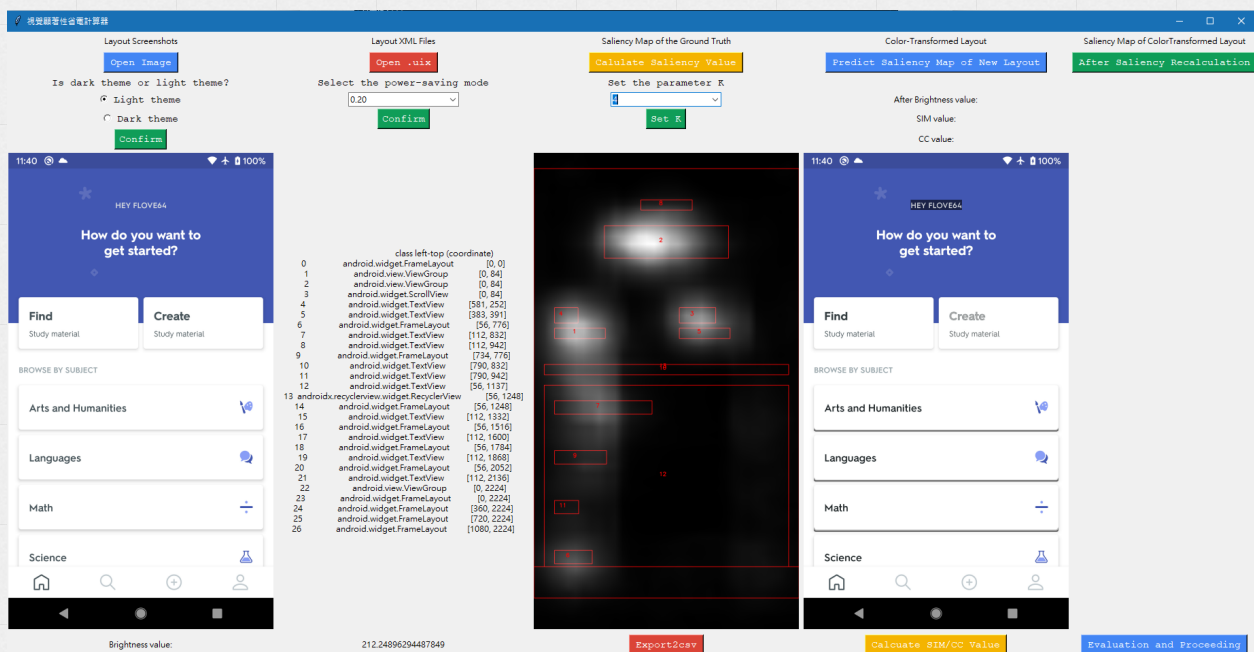
在計算期間，如果使用的是 Windows 系統，程式顯示無回應純屬正常現象，在 Windows 上對 Python 2 模型的調用導致該問題，可至終端介面查看計算過程：



完畢後，輸出元件範圍劃分：

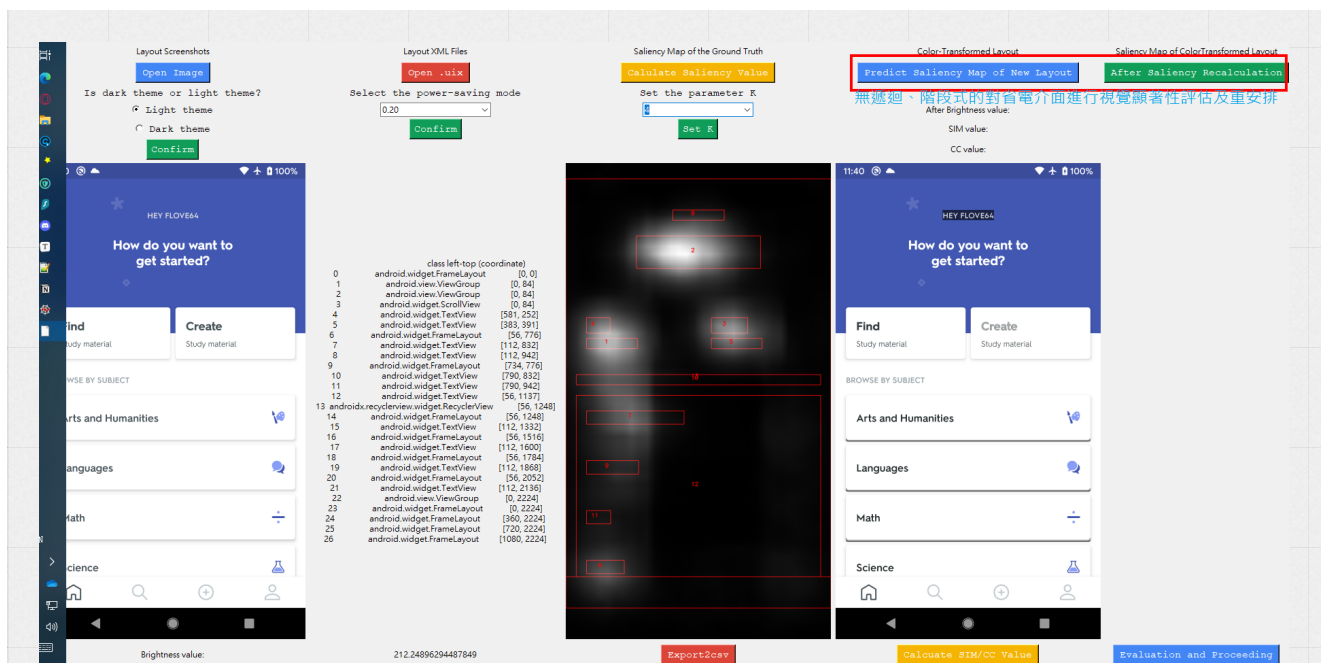


點擊確認後，會追加顯示元件排名，以及初始省電介面預覽：

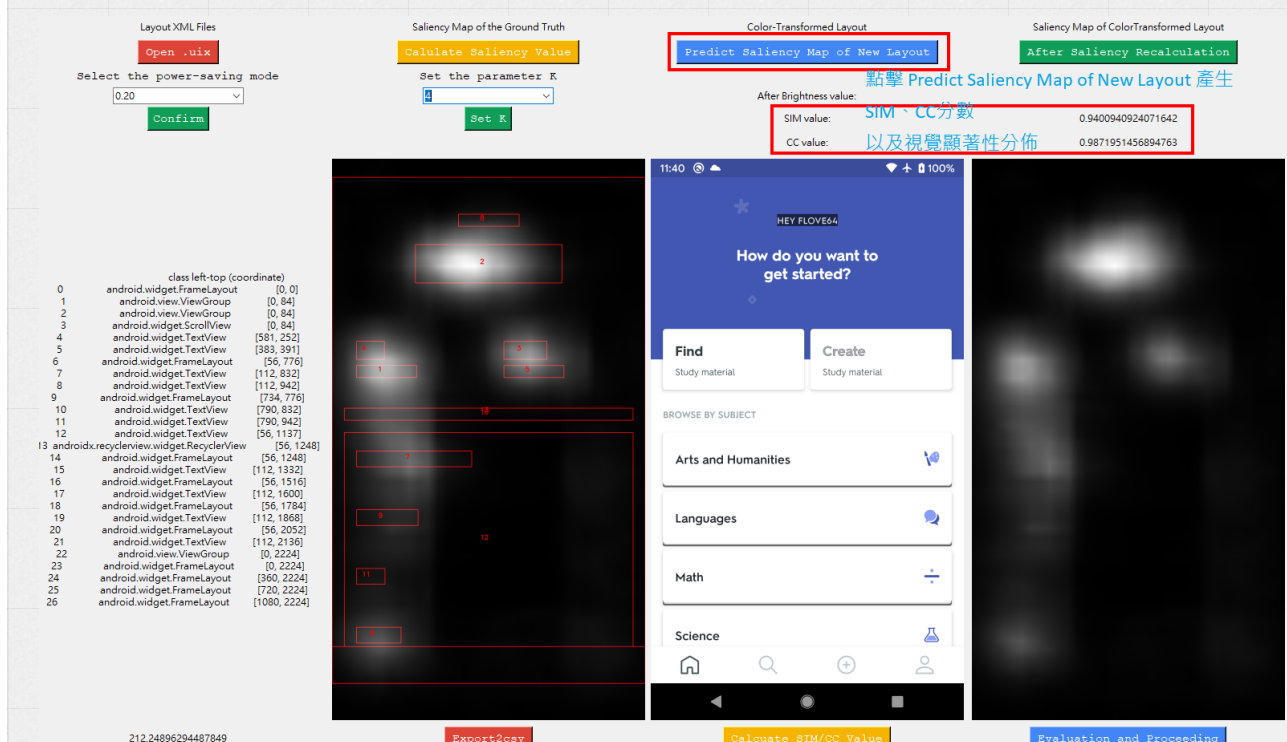


H3 測試是否能評估省電介面的視覺顯著性變化

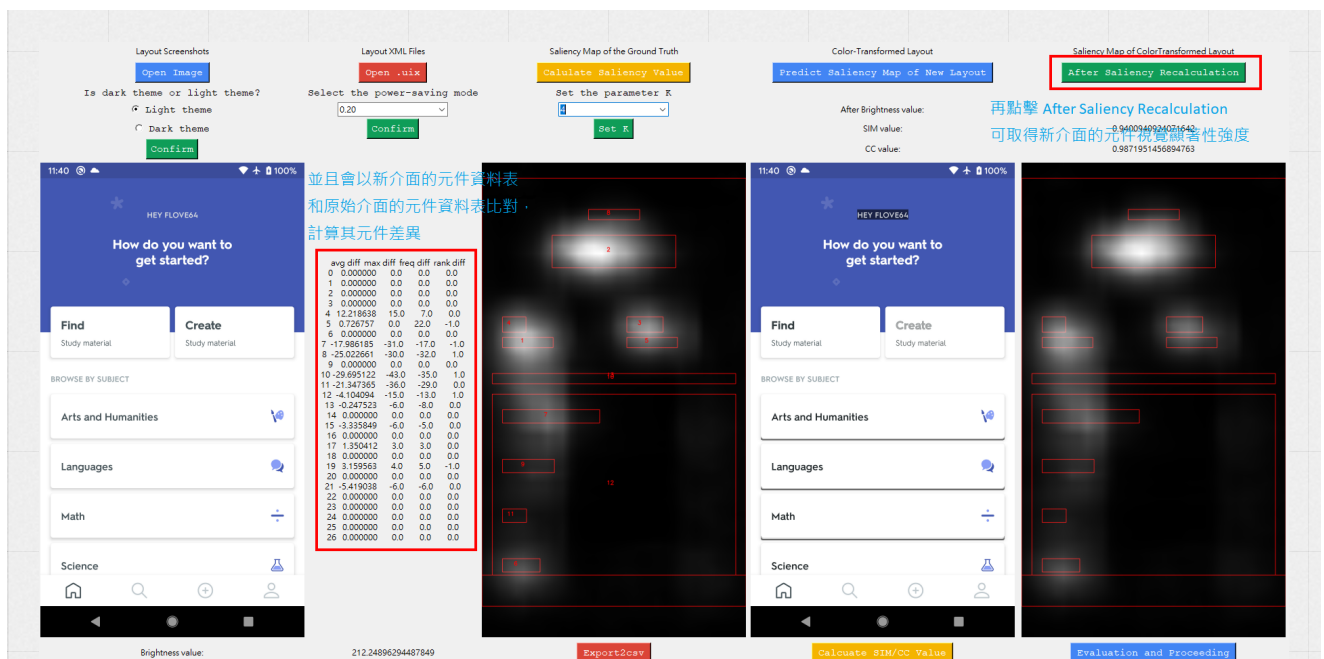
透過下列的操作方式，可以測試輸出省電介面的視覺顯著性分佈與差異評估：



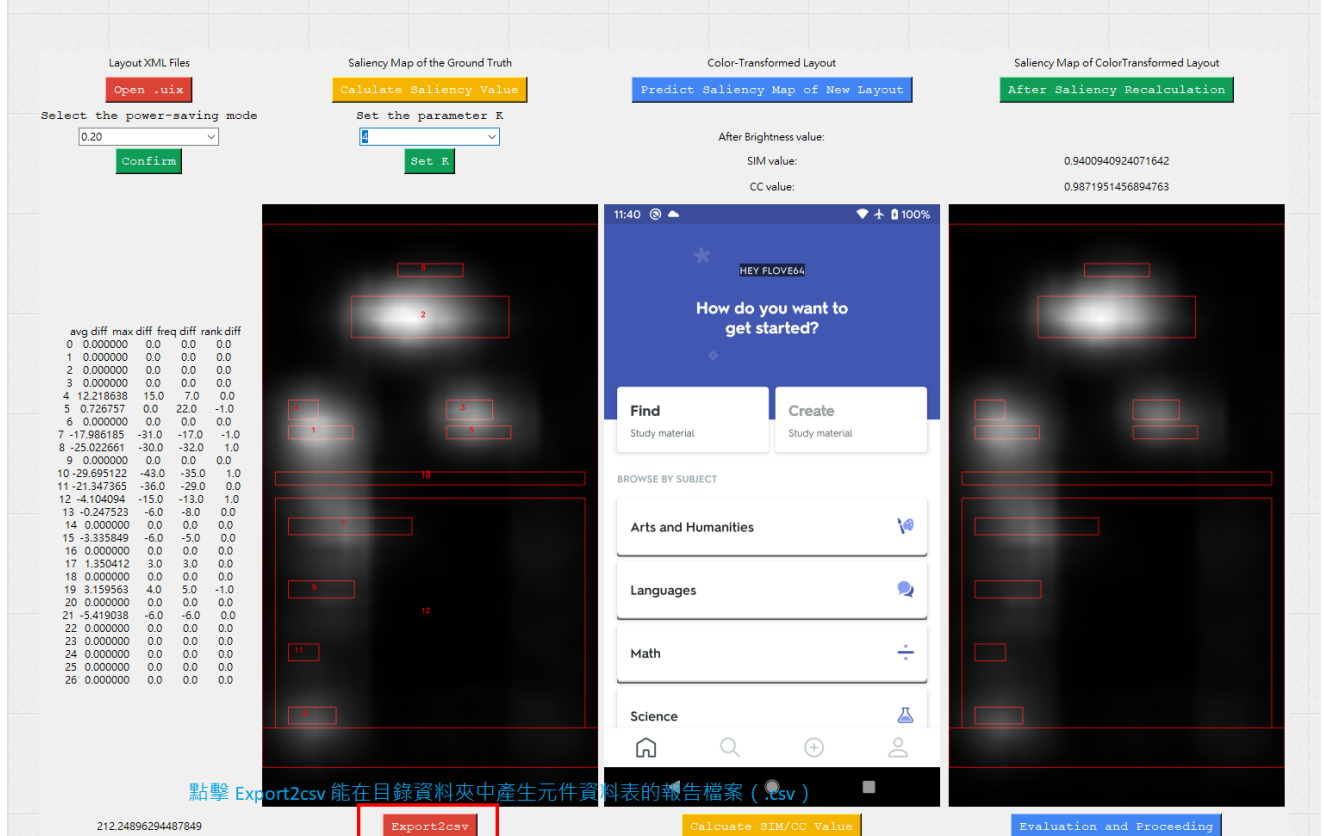
首先，點擊 Predict Saliency Map of New Layout，可於約30秒等待後，取得 SIM、CC分數，以及最右方產生預測的視覺顯著性分佈：



再點擊 After Saliency Recalculation，可以取得新舊介面的元件視覺顯著性差異（於元件資料表處顯示）：



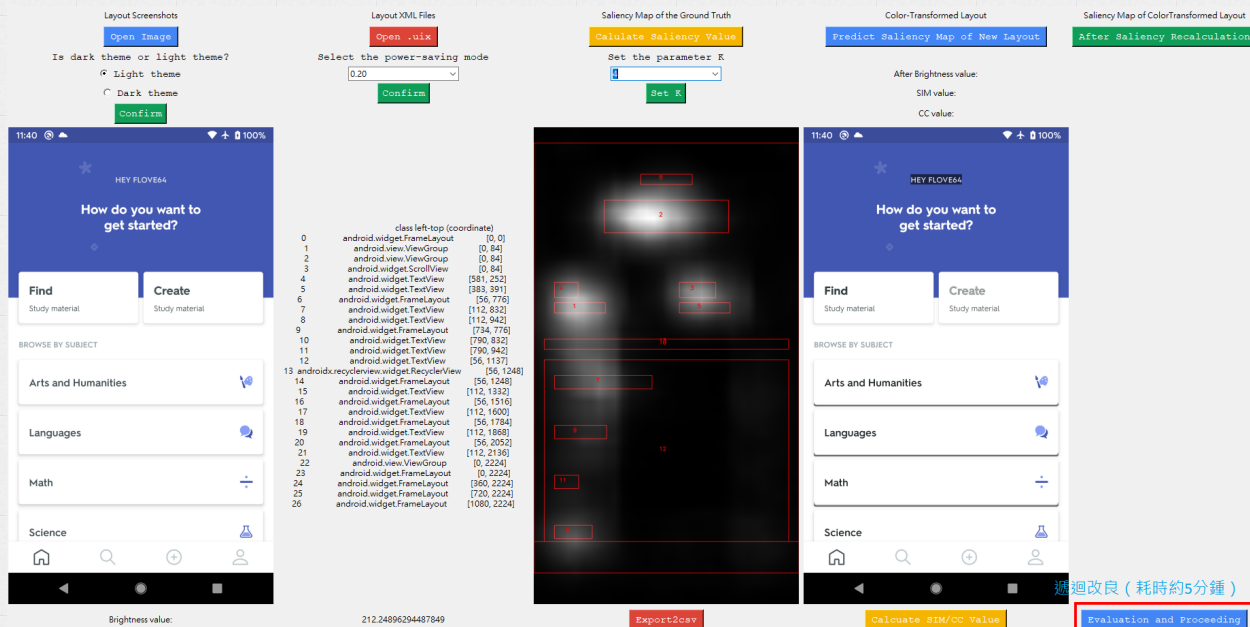
最終能透過點擊Export2csv，在目錄資料夾中產生報告檔案 (.csv)：



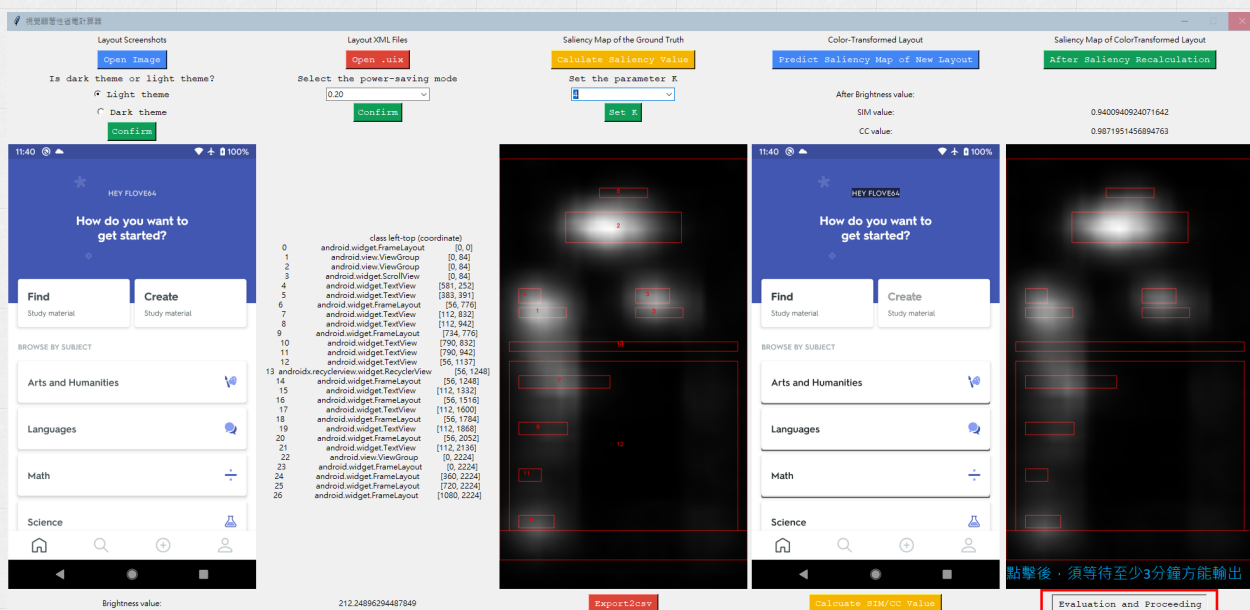
此步驟僅作為測試、除錯使用，若想進行遞迴改進省電介面，則需參考

H3 依照目標遞迴改進省電介面

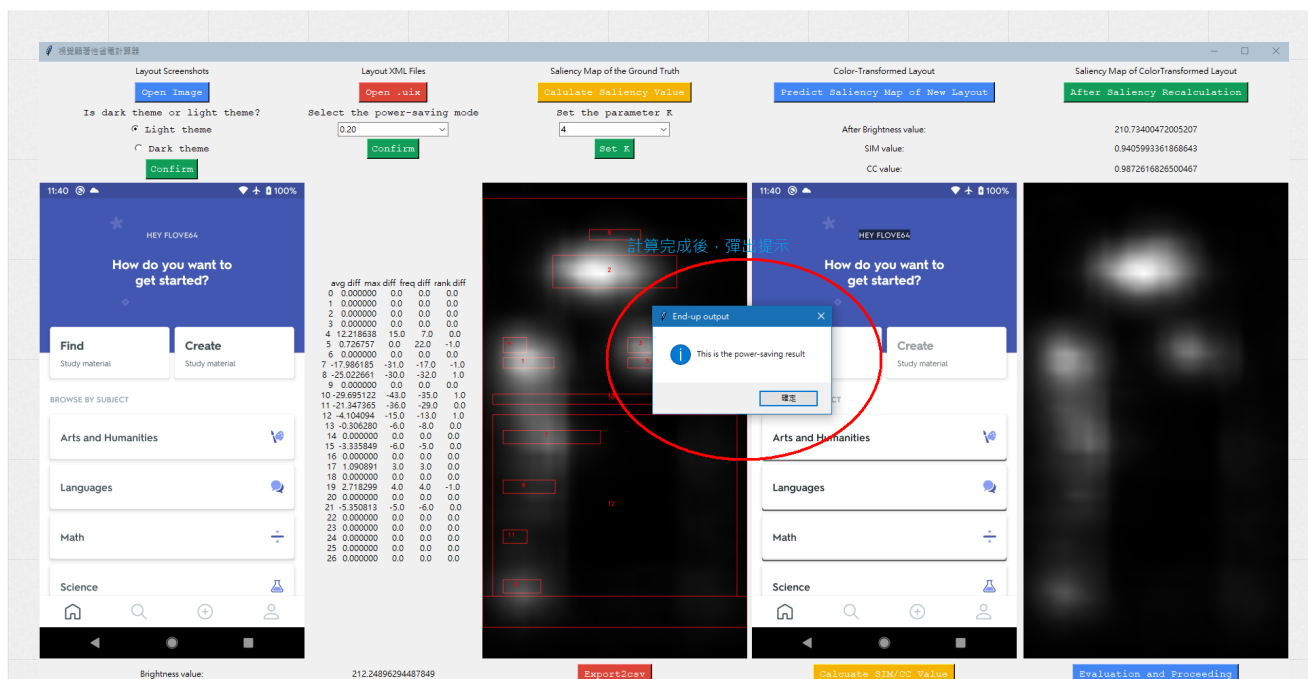
若前面測試時，功能均正常執行，則能直接執行右下方 Evaluation and Proceeding 按鈕功能，此功能整合前述3項功能，並輸出最終省電介面建議及色彩表：



點擊該按鈕後，通常需等待至少3分鐘，待程式計算完畢後，方有輸出（於Windows系統時，程式易進入無回應狀態）：



計算完成後，程式跳出提示：



H3 輸出檔案位置

程式會根據當初載入的介面圖片檔名，建立一個資料夾，內部生成的檔案即計算完成的檔案，其中

- newlayout.jpg資料夾存放省電版介面的預覽圖及其視覺顯著性分佈
- .csv檔案則是該介面的元件資料表及其色彩等級表



H3 後續運作

有了輸出色碼及省電介面預覽後，就可以至 Android Projects 中修改自己的應用程式介面，例如新增「省電版色彩」的切換按鈕，或是直接將程式介面遵照建議修改。